

ABSTRACT

VILLANES ARELLANO, ANDREA. An Analytical Approach for Bot Cheating Detection in a Massive Multiplayer Online Racing Game. (Under the direction of Michael Rappa).

The videogame industry is a growing business in the world, with an annual growth rate that exceeded 16.7% for the period 2005 through 2008. Moreover, revenues from online games will account for more than 38% of total video game software revenues by 2013. Due to this, online games are vulnerable to illicit player activity that results in cheating. Cheating in online games could damage the reputation of the game when honest players realize that their peers are cheating, resulting in the loss of trust from honest players, and ultimately reducing revenue for the game producers. Analysis of game data is fundamental for understanding player behaviors and combating cheating in online games. In this work, we propose a data analysis methodology to detect cheating in massively multiplayer online (MMO) racing games. More specifically, our work focuses on bot detection. A bot controls a player automatically and is characterized by repetitive behavior. Players in a MMO racing game can use bots to play during the races using artificial intelligence favoring their odds to win, and automate the process of starting a new race upon finishing the last one. This results in a high number of races played with race duration showing low mean and low standard deviation, and time in between races showing consistent low median value. A study case is built on upon data from a MMO racing game, and our results indicate that our methodology successfully characterize suspicious player behavior.

© Copyright 2012 by Andrea Villanes Arellano

All Rights Reserved

Analytical Approach for Bot Cheating Detection in a Massive Multiplayer Online Racing
Game

by
Andrea Villanes Arellano

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Master of Science

Computer Science

Raleigh, North Carolina

2013

APPROVED BY:

Aric La Barr

Benjamin Watson

Michael Rappa
Chair of Advisory Committee

DEDICATION

I would like to dedicate this work to my tia Mini for her immeasurable support, to my aunts (tia China, tia Jaco y tia Ne) for their unconditional love, and to my mom for her perseverance, patience and courage. To my family, because the road was easier with you in every corner. To my friends, for showing me that life is even more amazing when you start breaking your bubble.

BIOGRAPHY

Andrea Villanes Arellano was born in Lima, Peru in 1984. She received a Bachelor in Computer Science from the Pontificia Universidad Católica del Perú (PUCP), Lima, Perú. Upon receiving her bachelor degree, she joined North Carolina State University to pursue a M.S. in Analytics, which she obtained in May, 2009. In August 2009, she enrolled in the M.S. in Computer Science at North Carolina State University, where she began her research in the Data Science Lab under supervision of Dr. Rappa.

ACKNOWLEDGMENTS

First, I would like to acknowledge the members of my committee Dr. La Barr, Dr. Rappa and Dr. Watson for their guidance in this work. I would like to extend my special gratitude to Dr. Rappa for all his support and for being a great mentor to me.

I would like to extend my gratitude to Mark Cusick and Holle Christensen from the Data Science Lab for their ideas to this work, and to Oleg Veryovka for facilitating the data for this work, and providing us the gaming knowledge.

Finally, I would like to thank the department of Computer Science at North Carolina State University, all the staff and all my professors that taught me that Computer Science is more than I thought to know.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
1. INTRODUCTION	1
2. RELATED WORK	3
3. MATERIALS AND METHODS.....	8
3.1. Background Information	8
3.2. Data Description.....	9
3.3. Performance metrics.....	10
3.4. Data Analysis.....	10
4. RESULTS	17
5. DISCUSSION AND FUTURE WORK.....	37
REFERENCES	39

LIST OF TABLES

TABLE 1 - DESCRIPTION OF THE MOST RELEVANT DATA LOGGED IN THE GAME.....	9
TABLE 2 - RACE DURATION SUMMARY STATISTICS FOR THE EXCLUDED PLAYERS.....	18
TABLE 3 - CUTOFF VALUES FOR PVP, PVE AND PPVP FOR TIME IN BETWEEN RACES.....	19
TABLE 4 - CUTOFF VALUES FOR PVP, PVE AND PPVP FOR RACE DURATION.....	22
TABLE 5 - LISTING OF PVP POTENTIAL CHEATERS.....	28
TABLE 6 - SUMMARY TABLE FOR PVP POTENTIAL CHEATERS.....	29
TABLE 7 - TOP FIVE PVE POTENTIAL CHEATERS.....	29
TABLE 8 - BOTTOM FIVE PVE POTENTIAL CHEATERS.....	29
TABLE 9 - SUMMARY TABLE FOR PVE POTENTIAL CHEATERS.....	30
TABLE 10 - NODE 5: TOP FIVE PLAYERS.....	32
TABLE 11 - NODE 5: BOTTOM FIVE PLAYERS.....	32
TABLE 12 - SUMMARY TABLE OF NODE 5.....	33
TABLE 13 - TOP FIVE PPVP POTENTIAL CHEATERS.....	33
TABLE 14 - BOTTOM FIVE PPVP POTENTIAL CHEATERS.....	33
TABLE 15 - SUMMARY TABLE FOR PPVP POTENTIAL CHEATERS.....	34
TABLE 16 - NODE 3: LISTING OF PLAYERS.....	36
TABLE 17 - SUMMARY TABLE OF NODE 3.....	36

LIST OF FIGURES

FIGURE 1 - DATA ANALYSIS ALGORITHM.....	11
FIGURE 2 - DATA CLEANING ALGORITHM.....	12
FIGURE 3 - TIME IN BETWEEN RACES ALGORITHM.....	13
FIGURE 4 - RACE DURATION ALGORITHM.....	14
FIGURE 5 - MATCH PLAYERS ALGORITHM.....	15
FIGURE 6 - ANALYZE PLAYERS' RESULTS ALGORITHM.....	16
FIGURE 7 - SCATTER PLOT MW VERSUS SDW.....	18
FIGURE 8 - PVP: MEDIAN OF TIME IN BETWEEN RACES ANALYSIS.....	20
FIGURE 9 - PVE: MEDIAN OF TIME IN BETWEEN RACES ANALYSIS.....	21
FIGURE 10 - PPVP: MEDIAN OF TIME IN BETWEEN RACES ANALYSIS.....	21
FIGURE 11 - PVP: RACE DURATION ANALYSIS.....	23
FIGURE 12 - PVE: RACE DURATION ANALYSIS.....	24
FIGURE 13 - PPVP: RACE DURATION ANALYSIS.....	25
FIGURE 14 - PVP: MATCH PLAYERS RESULTS.....	26
FIGURE 15 - PVE: MATCH PLAYERS RESULTS.....	26
FIGURE 16 - PPVP: MATCH PLAYERS RESULTS.....	27
FIGURE 17 - SELECTING PLAYERS WITH WIN RATIO GREATER THAN 70%.....	28
FIGURE 18 - PVE POTENTIAL CHEATERS DECISION TREE.....	31
FIGURE 19 - PPVP POTENTIAL CHEATERS DECISION TREE.....	35

1. INTRODUCTION

The videogame industry generated nearly \$25 billion in revenue worldwide in 2011 [1]. According to the Entertainment Software Association (ESA), the annual sales in the U.S. were \$10.5 billion in 2009. Additionally, a strong growth in the video game software market has been seen during the past years, with an annual growth rate by the US game software industry that exceeded 16.7% for the period 2005 through 2008, while the growth for the U.S. economy was 2.8% during the same period [2]. Strategy Analytics, a leader in game reporting, reports that online games revenues will account for more than 38% of total video game software revenues by 2013 [3].

Massive multiplayer online games (MMOGs) continue to be a popular sector within the videogame industry. MMOGs support human players competing against others in a virtual world. These virtual worlds are persistent and material worlds, having in most cases a virtual currency that allows players to, for example, buy items that will enhance their gaming abilities [4]. It is in these virtual worlds where cheating exists, but not unchallenged.

Cheating in MMOGs is defined as [5]: “Any behavior that a player uses to gain an advantage over his peer players or achieve a target in an online game is cheating if, according to the game rules or at discretion of the game operator (that is, the game service provider, who is not necessarily the developer of the game), the advantage or the target is one that the player is not supposed to have achieved”.

Cheating in MMOGs represents a security concern for computer game designers because it affects the trust that honest players have in the game, which reduce their satisfaction on

playing the game, which ultimately reduces revenue for the game producers. Cheating in MMOGs is of several types: automation (bots), third-party software that directly affects the game play or modifies the results, and bugs in the game, which creates exploitable loopholes. Researchers have used machine learning techniques in order to detect cheating and understand players' behavior in online games. In this work, we study a MMO racing game and analyze race performance data collected over a period of six months to identify suspicious players. A suspicious player is defined as one that might be using additional resources other than his abilities, to win the game. The purpose of this work is to develop a set of guidelines that will serve identifying cheaters in online gaming, who are using bots to gain unfair advantage over other players.

2. RELATED WORK

This section explores previous work of defining and detecting cheating behavior in MMOGs. First, we begin by introducing the taxonomy created by Yan and Randell [5] that allow us to understand common forms of cheating in online games. Next, we present several works in literature that investigate how to detect or prevent cheating in online games.

Yan and Randell [5] define common forms of cheating in online games, and build a taxonomy:

- Exploiting misplaced trust: occurs when the cheater modifies the game client or data, so they can then replace the old copy and modify the behavior of the game. The modification of the game client or the data can also occur on the fly. This type of cheating happens because the developers of the game place too much trust in the client side.
- Collusion: occurs when players conspire with each other to gain advantage over other players.
- Abusing game procedure: occurs when the player abuses the games' operating procedure. In this type of cheating there is no need for technical expertise. For example, the player will disconnect from the game when they notice they are about to lose.
- Cheating related to virtual assets: occurs when players exchange real money with virtual money, and the player who is offering virtual money never delivers it after receiving the real money.

- Exploiting machine intelligence: occurs when the player uses artificial intelligence (AI) techniques that can compete with human players.
- Modifying client infrastructure: occurs when the player does not modify the client program or data on the client side, but they modify the client infrastructure. A popular cheating technique, called wall-hacking, allows the player to modify the drivers on their computer and make a wall transparent to give them advantage over their peers.
- Denying service to peer players: occurs when players deny service to their peer players. A common technique will overflow the opponent's network connection to delay their actions.
- Timing cheating: occurs when players delay their actions until they know everyone else's actions.
- Compromising passwords: occurs when players gain access to other players' password, which allow the cheater to access all the player's information and virtual assets.
- Exploiting lack of secrecy: occurs when game programs exchange communication packets in plaintext format. The cheater can then insert, delete or modify information in the packets.
- Exploiting lack of authentication: occurs when there does not exist a proper mechanism of authentication. This type of cheating is more common in countries like China and Korea where players can physically access different computers and access other players' information.

- Exploiting a bug or loophole: occurs when cheaters exploit a game defect in the game program or in the game design.
- Compromising game servers: occurs when players modify game server programs or change configurations in the game host system.
- Internal misuse: occurs when insiders abuse privileges in the game when they have system administrator privileges.
- Social engineering: occurs when cheaters trick honest players making them believe that they need to re-enter their ID and password in another website.

There has also been research that combines heuristic or machine learning based anomaly detection in order to detect cheating in online games. Laurens et. al. [6] presented a real-time method to detect cheating behavior, specifically wall-hacking, which monitors player behavior for indications of cheating play. Wall-hacking, in which the player makes the wall transparent, allows the player to see through the walls and anticipates other players' moves. The method monitors players' movement, and starts with the hypothesis that cheating behavior is significantly distinguishable from normal play. The system was developed for a single type of cheat, wall-hacking, in a first-person-shooter (FPS) game. The system keeps tracks on the traces of each player. A trace is defined at what the player is 'looking' at, at any moment in time, and what that object in the world is. An illegal trace is a trace from the player to an enemy where the player's line of sight passes through an opaque world material before it reaches the enemy. The system defines four metrics to detect illegal behavior: a) frequency of illegal traces; b) sequence of illegal traces; c) distance to world traces; d) and

distance to entity traces. A final ‘cheat-score’, which combined metrics, was used to calculate which players were mostly likely to be cheating in the game.

Galli et. al. [7] introduced a framework to automatically detect cheating behaviors in Unreal Tournament III, a FPS game, by using several supervised learning techniques, i.e., decision trees, Naïve Bayes, random forest, neural networks and support vector machines. The authors developed an experimental cheating system, which allowed them to record and process incoming data. The incoming data provided a very low-level description of the players’ actions, and it was labeled as a ‘cheater’ if the player used a cheating system during that action. The incoming data was used to build a decision model from a labeled dataset, and apply the model to classify an unlabeled dataset using five methods of supervised learning techniques: decision trees, Naïve Bayes, random forest, neural networks and support vector machines. As a result, they were able to correctly classify almost 90% of the test examples. Although work has been done in FPS games, Chapel et. al. [8] proposed a probabilistic framework to identify cheating players based on their playing behavior. The authors assume that the player has a rank, which determines the probability of the outcomes of their games. Their hypothesis is that players that are cheating will have an artificially inflated rank. The rank does not correspond to any in-game ranking, but it is estimated from the game results. They constructed a probabilistic model for the result of the games between different players and detected potential cheaters based on statistical tests.

Work has also been done in bot detection. A bot controls a player automatically, with or without artificial intelligence, and allows the player to gain rewards without making the efforts required by the game [9]. Kim et. al. [10] proposed a method where event sequences

were analyzed to detect players using auto programs in Massively Multi-player Online Role Playing Games (MMORPGs). Auto programs are software or hardware that are used on behalf of human players to gain advantage over honest players. Auto programs produce automatic keyboard or mouse events that replace human actions. However, auto programs harm games servers in several ways, including abuse of resources. The method proposed analyzes the event sequences to detect which players are using auto programs. The event sequences are transformed into a set of attributes, and a Decision Tree is used to classify the cheaters. The results show high accuracy to detect auto programs. Pao et. al. [9] proposed a method to automatically detect the use of game bots in online games. Their method consists of detecting bots based on the dissimilarity measurements between the trajectories of a bot and a human player. Their method focuses on games where avatars' movements are directly controlled by players, such as FPS games. The method takes as the input the avatars' movement trajectory, and then applies a supervised learning algorithm to classify which trajectory was made by a human or by a bot.

Our work differs on the previous work in that this is the first work done in detecting bots in a MMO racing game. More specifically, this work uses data mining in order to detect suspicious players who might be using bots to gain unfair advantage over their peers. A methodology is proposed in order to analyze data collected from an online racing game, and it results in a set of potential cheaters and their data characterization.

3. MATERIALS AND METHODS

3.1. Background Information

The data for this work comes from a racing multiplayer online game. The game contains around 2.5 million registered users. A user has a unique account identifier. A user can have multiple cars, and multiple drivers. The game consists of a series of tracks. Tracks in the game are unlocked as the player levels up. When the player levels up, they earn money, which they can use to buy higher performance cars. Players can also use real world currency to buy higher performance cars regardless of their level in the game. The player selects a track to race and a racing mode. A player can have multiple sessions, and multiple races within each session. A session is defined to start at the time when the player signs in their account, and ends when the player logs out of their account. Three racing modes are supported by the game:

- Player vs. player (PVP): players play against other. The number of players in a race is limited to seven.
- Player vs. environment (PVE): the player plays against artificial intelligent players (environment).
- Private player vs. player (PPVP): players invite their friends to play in a private race.

Each mode presents different characteristics that may influence some of the performance metrics (i.e. time in between races). For example, the PVE mode is more deterministic than any of the other two modes because the player is competing against the environment.

A player suspected of being a cheater may show certain characteristics. It is our goal to use performance metrics to uncover patterns of behavior that should help in identifying suspicious players. Time in between sessions would not be an indication of a player using bots, but a consistent time in between races would be. Being a winner is not, but winning a disproportionate number of races is. Participating frequently in the races is not, but participating in a larger than expected number of races may also be symptomatic of suspicious players.

3.2. Data Description

The dataset in this work contained a total of 2.2 million observations for 45,457 distinct players. Each observation in the dataset is a race instance. The dataset contained the following variables as shown in Table 1.

Table 1 - Description of the most relevant data logged in the game

Name	Range	Description
LocalUserID	-	Unique identifier of a user
EventID	-	Unique ID of a race
ModeID	{1, 2, 3}	ID of race mode: PVP, PVE, Private PVP
EventSessionID	-	ID of a particular race session
ResultValue	{1, 2, 3, 4, 5, 6, 7}	Placing in the race 1-7. 1 is a win
EventStartDtm	20JUL10:00:03:46 - 02FEB11:00:33:14	Race start time
EventEndDtm	20JUL10:00:05:56.910 - 02FEB11:00:34:35.750	Race end time
EventDuration	0 - 4294960 seconds	Race duration

3.3. Performance metrics

Time in between races: the time it takes a player to start another race upon finishing up his previous race.

Race duration: the time it takes a player to finish a race. Race durations vary depending on the track the player is playing.

3.4. Data Analysis

The objective of the analysis is to discover suspicious players who are cheating in the game using performance data collected by online gaming companies. Players who might be using bots are of particular interest in this work. We propose the following steps to identify potentially suspicious players: the time in between races, race duration and number of races played are used to differentiate between humans and bots. This can be done because of the consistent repetitive movement that characterizes bots. The following statistics are calculated: (a) median of time in between races, (b) number of total races played by player, (c) number of races played by track and player, (d) mean of race duration by track and player, and (e) standard deviation of race duration by track and player. A “bot player” will present the following characteristics: (a) low median of the time in between races, (b) high number of total races played, (c) high number of races played by track (d) low mean of race duration by track, and (e) consistent race durations by track. Figure 1 presents succinctly the proposed steps to uncover potential suspicious players followed by their description.

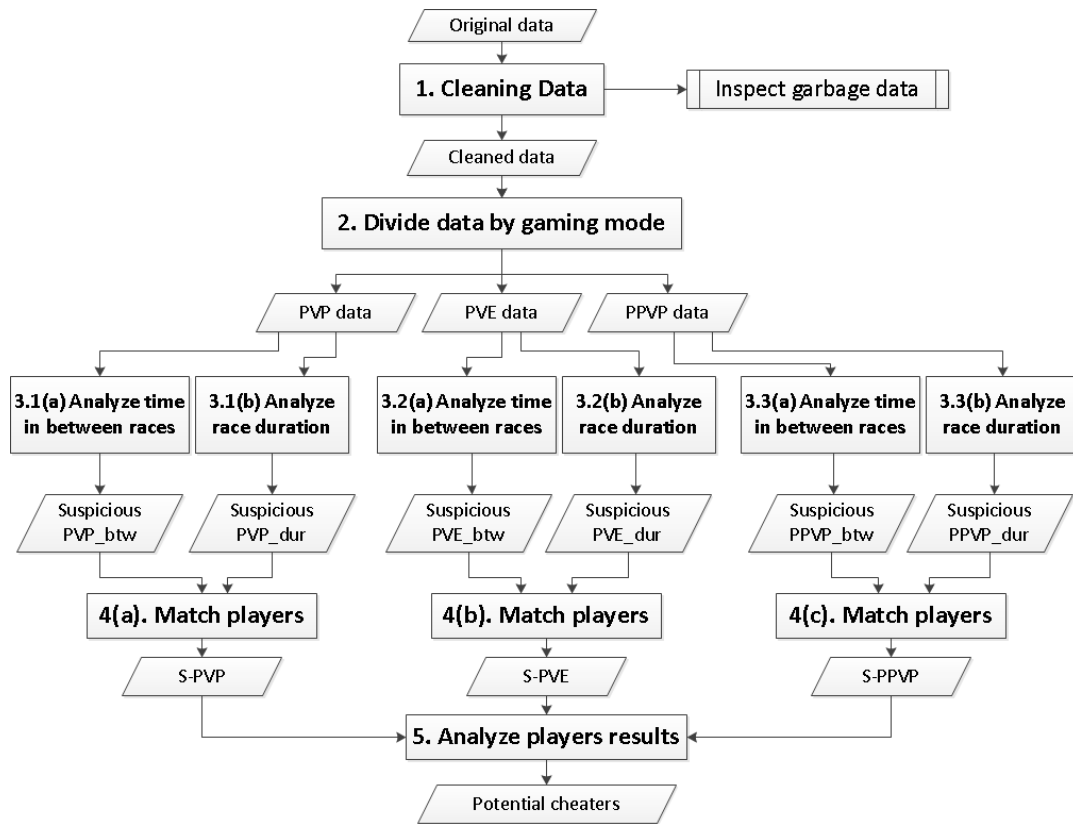


Figure 1 – Data analysis algorithm

1. Cleaning the data: the purpose of this step is to eliminate noisy data not useful for the analysis. Mean and standard deviation of race duration by player are calculated over all tracks and races. Data points close to the origin (0,0) are of particular interest because these points represent “unusually fast” players with low standard deviation and low mean. These players are either obvious cheaters with all races ending within a few seconds or garbage data. This data is inspected separately before eliminating it from the analysis in order to make sure that the data is not of value for our purposes. In order to avoid eliminating potential cheaters, the number of races played is

calculated. Players with a low number of races (i.e. less than five races) are not suspicious of being cheaters, but a player with a high number of races would be, and should not be eliminated as garbage data. Figure 2 presents the data cleaning algorithm.

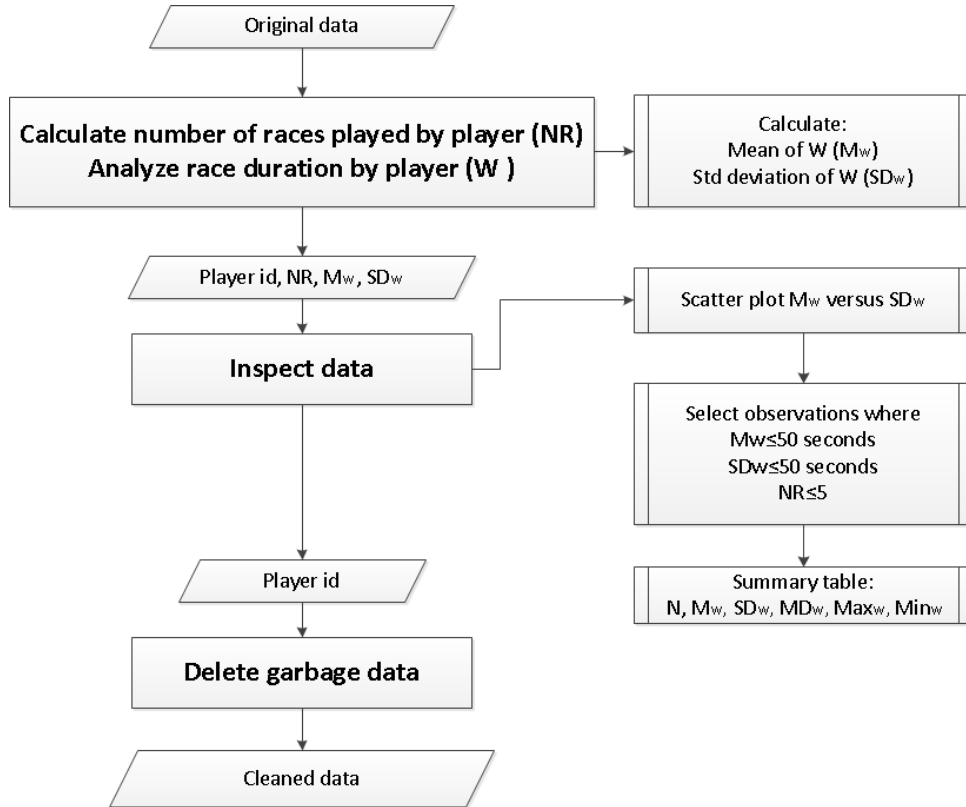


Figure 2 – Data Cleaning algorithm

2. Analyze the time in between races: the purpose of this step is to uncover suspicious players who might be using bots in order to automate the time in between races. These would be players who had low time in between races, and high number of races played. Two cutoff values: k for the median of the time in between races, and v for

the number of races played are chosen in this step in order to identify suspicious players. Suspicious players are the ones who had a median of the time in between races less than k , and a number of races played greater than v . Figure 3 presents the time in between races algorithm.

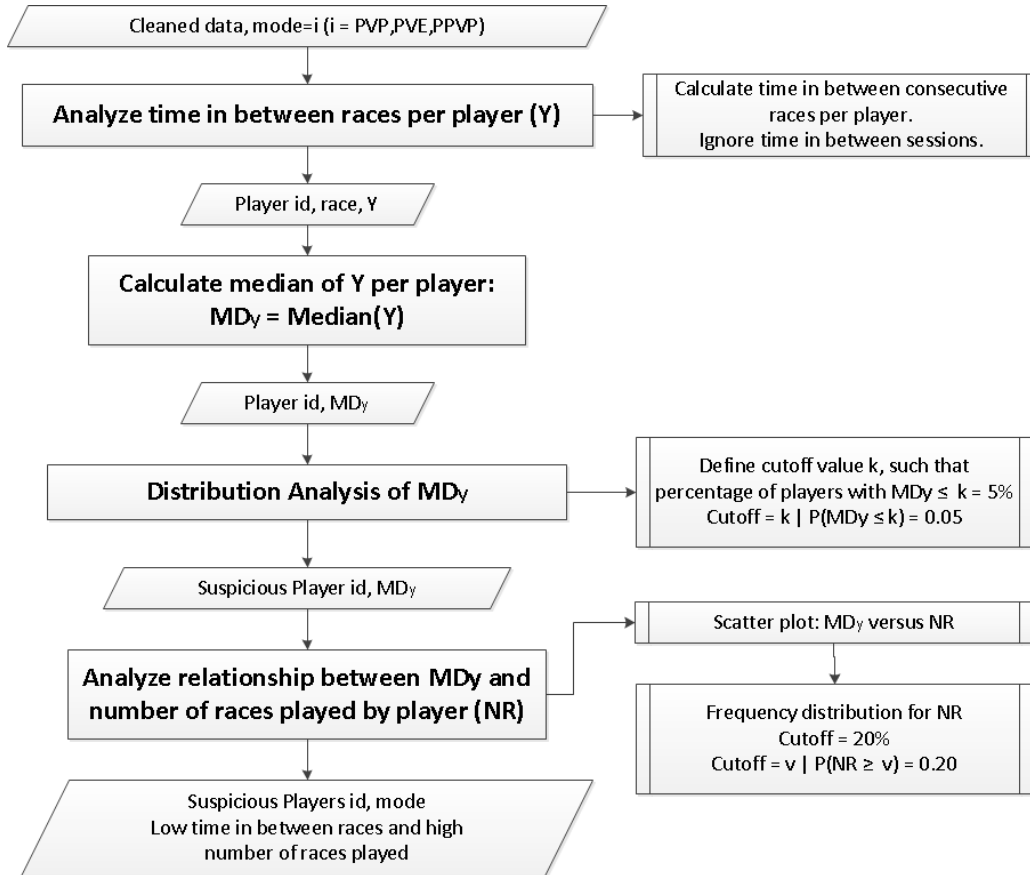


Figure 3 – Time in between races algorithm

3. Analyze the race duration: the purpose of this step is to uncover players who might be using bots during the races. These would be players who completed the races in a short time, and show the same times consistently across the races in the same track. Three cutoff values: j for the number of races played by player and track, p for the mean race duration and q for the standard deviation of race duration are selected in this step in order to identify suspicious players in this step. Suspicious players are the ones who had a mean race duration lower than p , a standard deviation for race duration lower than q , and a number of races played by track greater than j . Figure 4 presents the race duration algorithm.

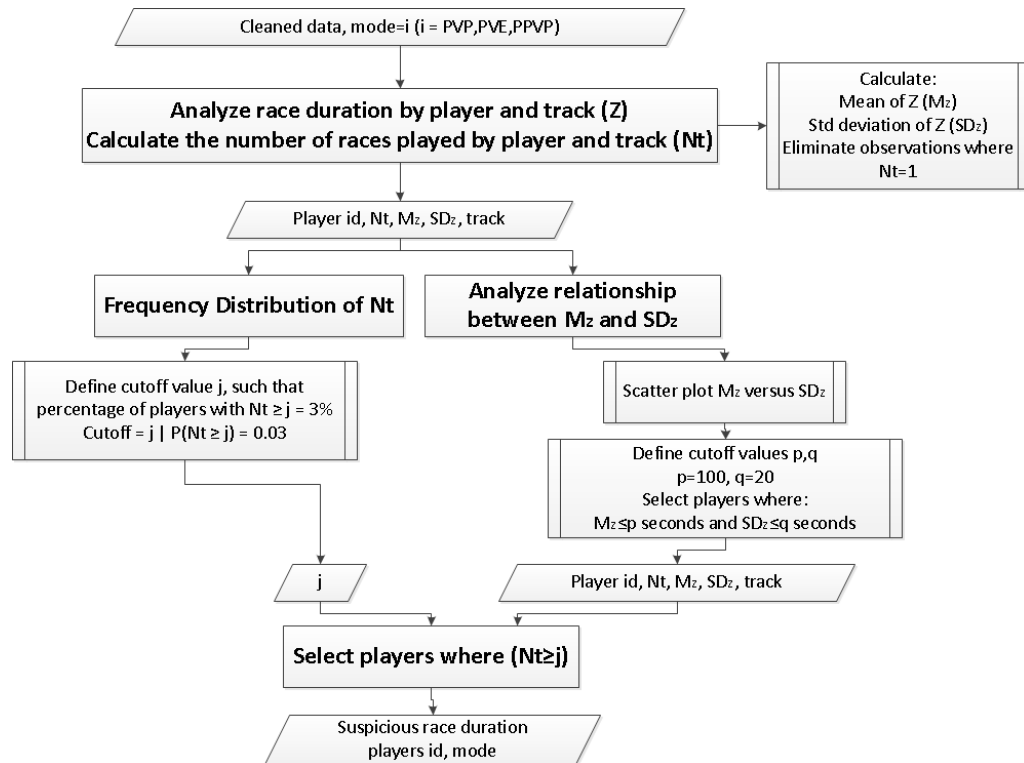


Figure 4 – Race duration algorithm

4. Match players: the two datasets from steps 2 and 3 are matched by the player id to find out players present in both datasets. Figure 5 presents the match players algorithm.

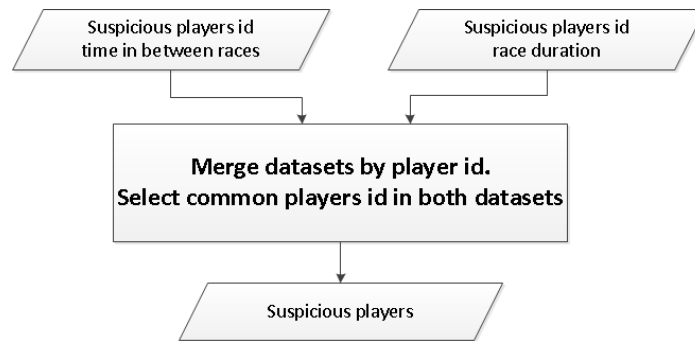


Figure 5 – Match players algorithm

5. Analyze players' results: the purpose of this step is to characterize potential cheaters, and analyze their placing behavior in the game through summary statistics and expert defined activity rules. Figure 6 presents the analyze players' results algorithm.

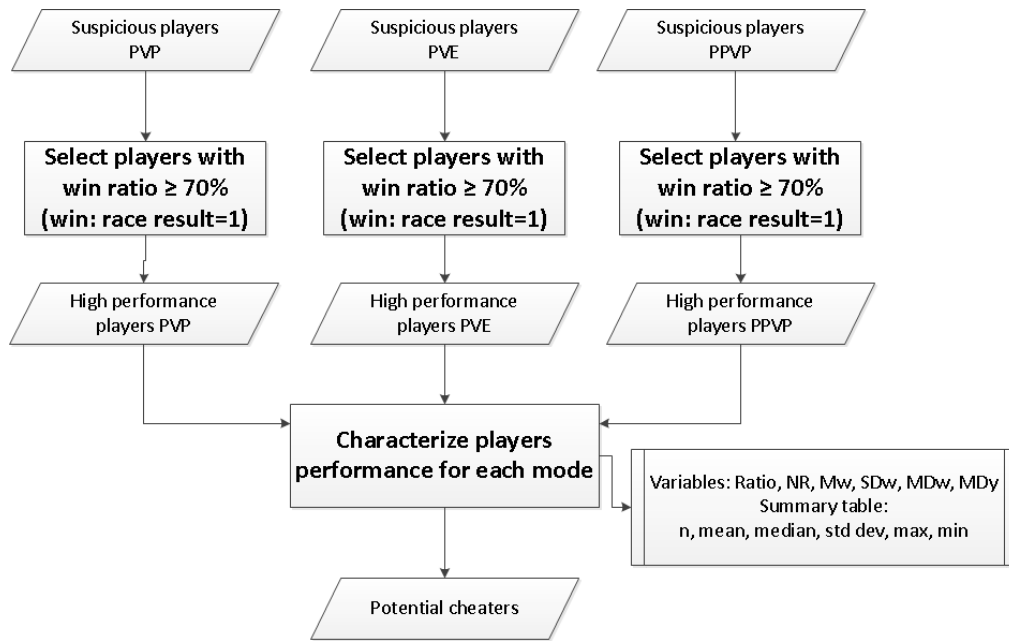


Figure 6 – Analyze players' results algorithm

4. RESULTS

1. Cleaning the data

The first step in the main algorithm is to clean the data. The number of races played by a player is calculated, and a scatter plot of the mean versus the standard deviation of race duration by player (W) is plotted as shown in Figure 7. After examination, it was decided that noisy observations would correspond to races with a mean of race duration of less than 50 seconds, and a standard deviation of less than 50 seconds. These are players with smaller times and standard deviations indicating possibly a small number of races, and abandoning the race. Data points colored in red are players that have: (a) Mean of race duration < 50 seconds, (b) Standard deviation of race duration < 50 seconds.

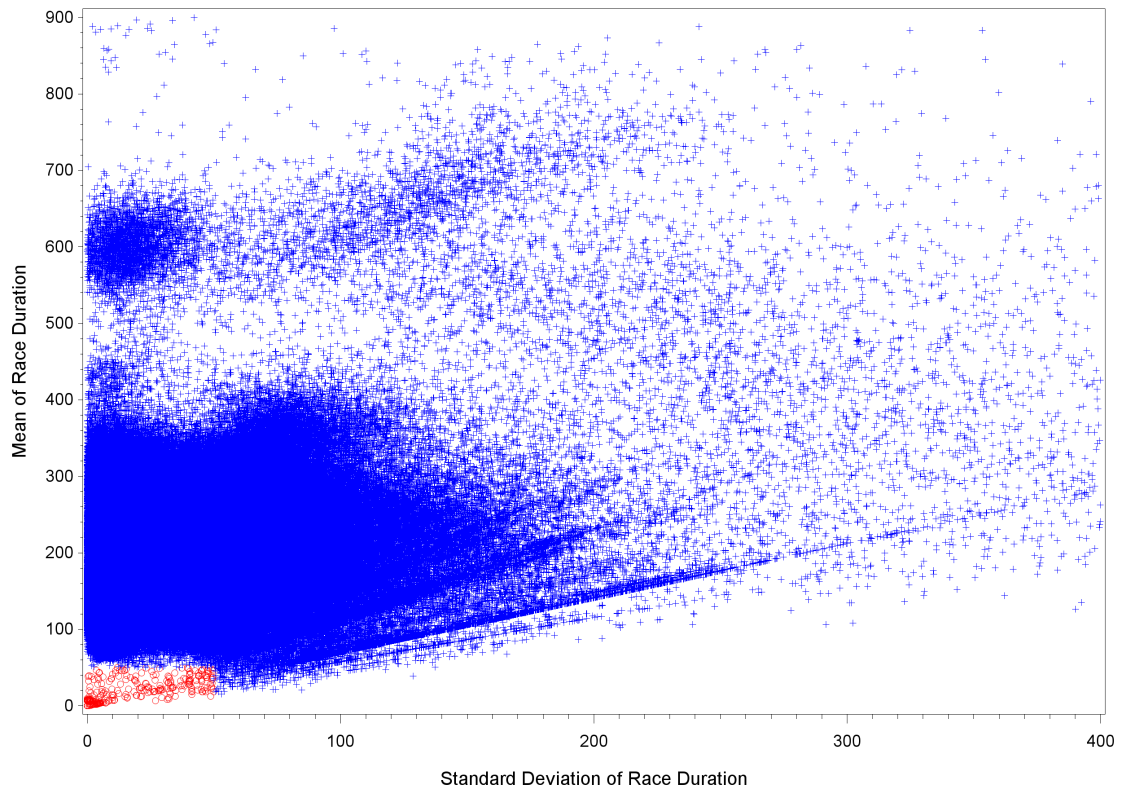


Figure 7 – Scatter plot Mw versus SDw

The number of races played by the players in the red cluster is inspected, concluding that 98.68% of the players had five or fewer races played in total. These players are eliminated from the dataset and excluded in the analysis. Table 2 shows race duration summary statistics for the 4478 players excluded from the analysis.

Table 2 – Race duration summary statistics for the excluded players

<i>statistic</i>	<i>n</i>	<i>Mean</i>	<i>Std dev</i>	<i>Median</i>	<i>Min</i>	<i>Max</i>
Mean (Mw)	4478	10.9793	15.3840	1.09200	0	49.9940
Std (SDw)	574	9.9332	15.6683	0.01732	0	49.9877

The data is divided by mode (i.e. PVP, PVE, PPVP) in order to continue with the algorithm.

2. Analyze the time in between races

The median (MDy) of the time in between races (Y) is calculated per player, and its distribution is analyzed. The cutoff value k for suspicious players is defined, such that:

$$\text{Percentage of players with MDy} \leq k = 5\%$$

$$\text{Cutoff} = k \mid P(\text{MDy} \leq k) = 0.05$$

A scatter plot of the MDy versus the number of races played by the player (NR) for those players with $\text{MDy} \leq k$ is analyzed. A second cutoff value v , for the number of races played by each player (NR), is defined for suspicious players, such that:

$$\text{Percentage of players with NR} \geq v = 20\%$$

$$\text{Cutoff} = v \mid P(\text{NR} \geq v) = 0.20$$

Players with a low median time in between races (MDy) and high number of races (NR) are selected as a suspicious group based on the time in between races.

The cutoff values for PVP, PVE and PPVP are shown in Table 3.

Table 3 – Cutoff values for PVP, PVE and PPVP for time in between races

<i>Cutoff value</i>	<i>PVP</i>	<i>PVE</i>	<i>PPVP</i>
k	90 seconds	40 seconds	104 seconds
v	2000 races	1000 races	400 races

a. PVP:

Figure 8 shows the distribution of MDy, and the scatter plot of NR vs. MDy. The red dotted line in the distribution of MDy indicates the cutoff value $k=90$, and the red dotted line in the scatter plot indicates the cutoff value for $v=2000$. A total of 213 suspicious PVP players based on time in between races are selected.

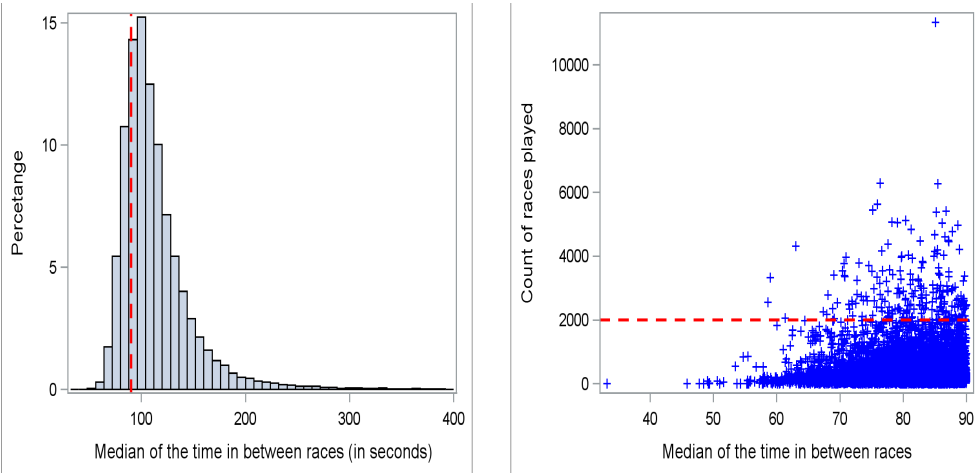


Figure 8 – PVP: Median of time in between races analysis

b. PVE

Figure 9 shows the distribution of MDy, and the scatter plot of NR vs. MDy. The red dotted line in the distribution of MDy indicates the cutoff value $k=40$, and the red dotted line in the scatter plot indicates the cutoff value for $v=1000$. A total of 284 suspicious PVE players based on time in between races are selected.

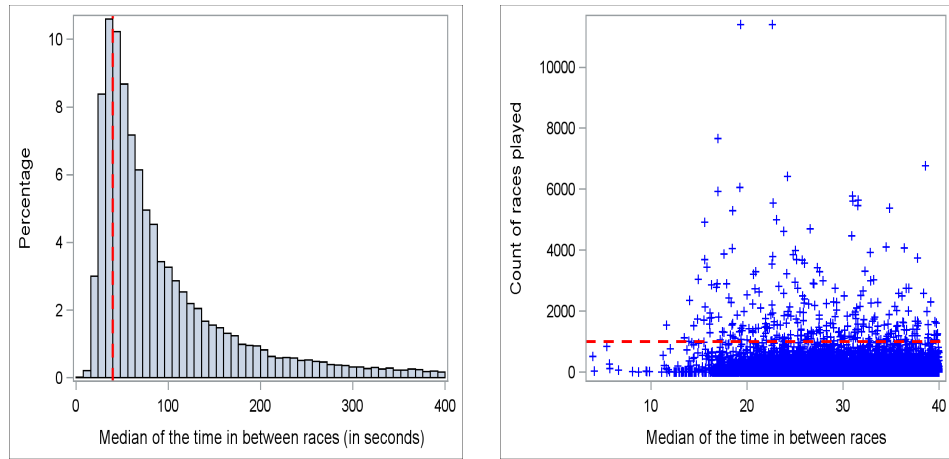


Figure 9 – PVE: Median of time in between races analysis

c. PPVP

Figure 10 shows the distribution of MDy, and the scatter plot of NR vs. MDy. The red dotted line in the distribution of MDy indicates the cutoff value $k=104$, and the red dotted line in the scatter plot indicates the cutoff value for $v=40$. A total of 315 suspicious PPVP players based on time in between races are selected.

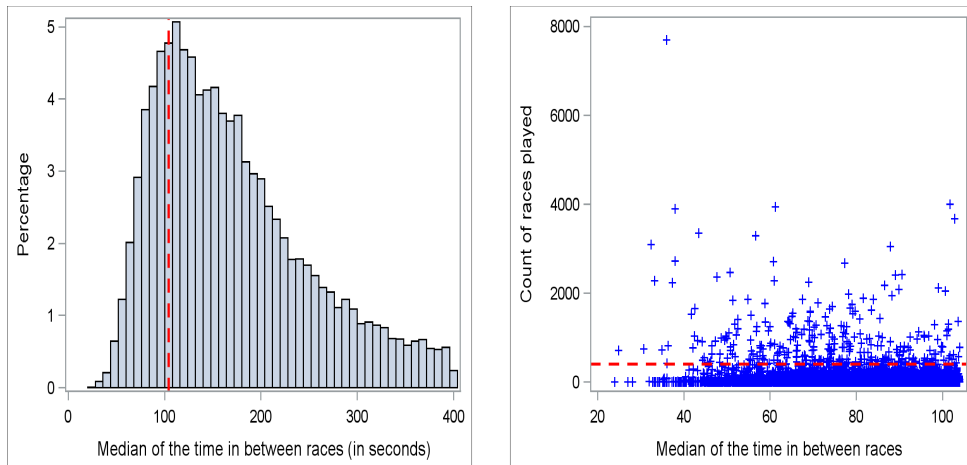


Figure 10 – PPVP: Median of time in between races analysis

3. Analyze race duration

The number of races played by player and track (N_t), the mean of the race duration by player and track (M_z), and the standard deviation of the race duration by player and track (SD_z) are calculated. Observations where $N_t=1$ are eliminated because these observations have $SD_z=0$.

A cutoff value j for N_t is defined, such that:

$$\text{Percentage of players with } N_t \geq j = 3\%$$

$$\text{Cutoff} = j \mid P(N_t \geq j) = 0.03$$

A scatter plot of M_z versus SD_z is analyzed. Players with the following characteristics are selected as suspicious: a) $N_t \geq j$, b) $M_z \leq 100$ seconds, and c) $SD_z \leq 20$ seconds.

Players with a mean of race duration per track (M_z) lower or equal than 100 seconds, standard deviation of race duration per track (SD_z) lower or equal than 20 seconds, and a number of races per track (N_t) greater or equal than j , depending on the mode, are selected as a suspicious group based on race duration.

The cutoff values for PVP, PVE and PPVP are shown in Table 4.

Table 4 – Cutoff values for PVP, PVE and PPVP for race duration

<i>Cutoff value</i>	<i>PVP</i>	<i>PVE</i>	<i>PPVP</i>
j	100 races	50 races	30 races
M_z	100 s	100 s	100 s
SD_z	20 s	20 s	20 s

a. PVP

Figure 11 shows the distribution of N_t , and the scatter plot of M_z vs. SD_z . The red dotted line in the distribution of N_t indicates the cutoff value $j=100$, and the red cluster close to the origin indicates those players with a $M_z \leq 100$ and a $SD \leq 20$. Within the red cluster, the players who had a $N_t > 100$ are selected. A total of 6233 suspicious PVP players based on race duration are selected.

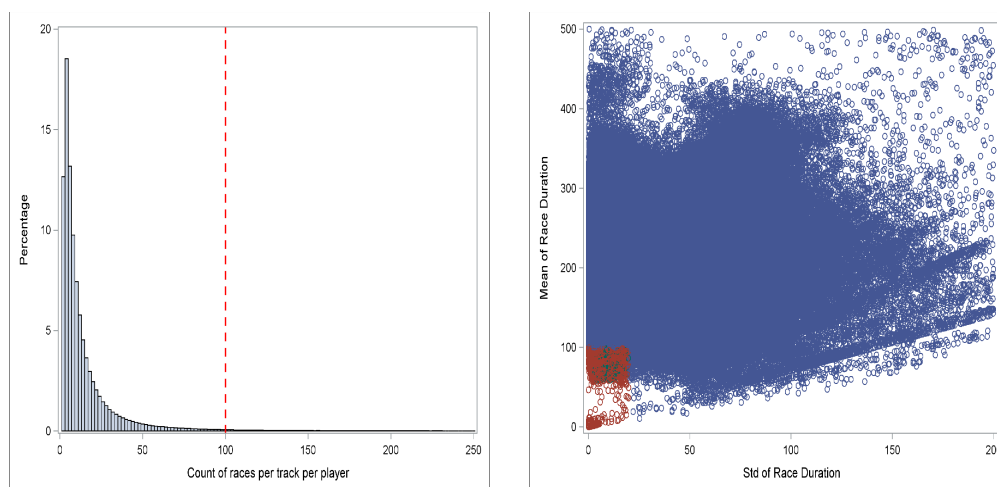


Figure 11 – PVP: Race duration analysis

b. PVE

Figure 12 shows the distribution of N_t , and the scatter plot of M_z vs. SD_z . The red dotted line in the distribution of N_t indicates the cutoff value $j=50$, and the red cluster close to the origin indicates those players with a $M_z \leq 100$ and a $SD \leq 20$. Within the red cluster, the players who had a $N_t > 50$ are selected. A total of 1515 suspicious PVE players based on race duration are selected.

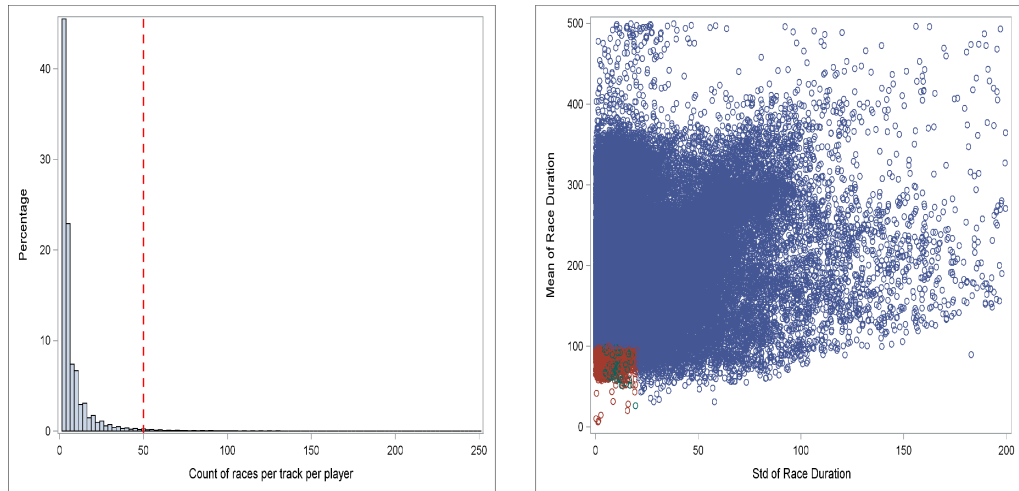


Figure 12 – PVE: Race duration analysis

c. PPVP

Figure 13 shows the distribution of N_t , and the scatter plot of M_z vs. SD_z . The red dotted line in the distribution of N_t indicates the cutoff value $j=30$, and the red cluster close to the origin indicates those players with a $M_z \leq 100$ and a $SD \leq 20$. Within the red cluster, the players who had a $N_t > 30$ are selected. A total of 1201 suspicious PPVP players based on race duration are selected.

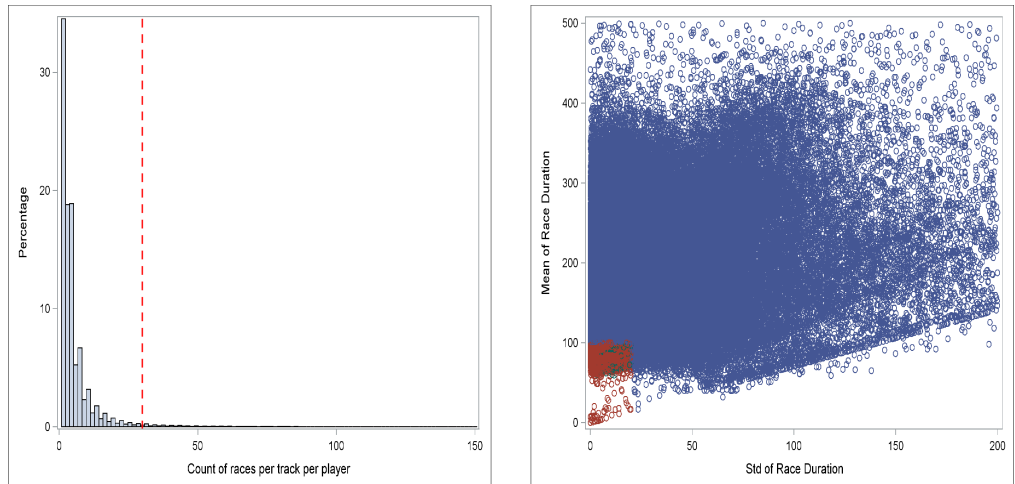


Figure 13 – PPVP: Race duration analysis

4. Match players

The two datasets from steps 2 and 3 are matched by the player id to find out suspicious players present in both datasets. These would be players who had low time in between races, high number of races played, players who completed the races in a short time, and showed the same times consistently across the races in the same track. This step is performed by mode.

a. PVP

A dataset with a total of 208 PVP suspicious players is obtained after merging the two datasets from the previous steps as shown in Figure 14.

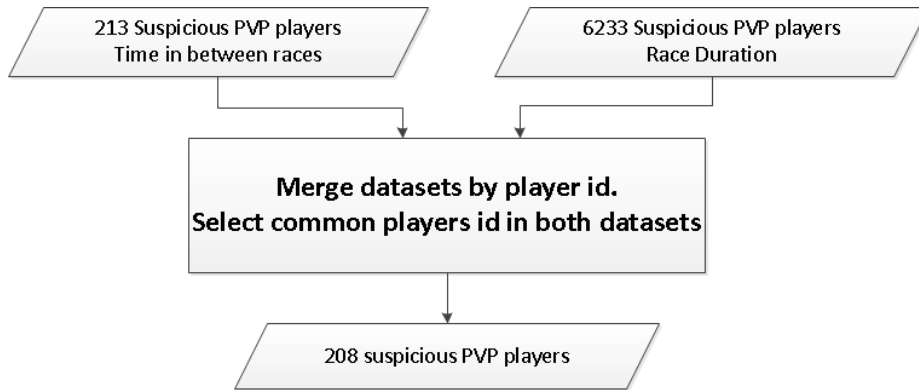


Figure 14 – PVP: Match players results

b. PVE

A dataset with a total of 214 PVE suspicious players is obtained after merging the two datasets from the previous steps as shown in Figure 15.

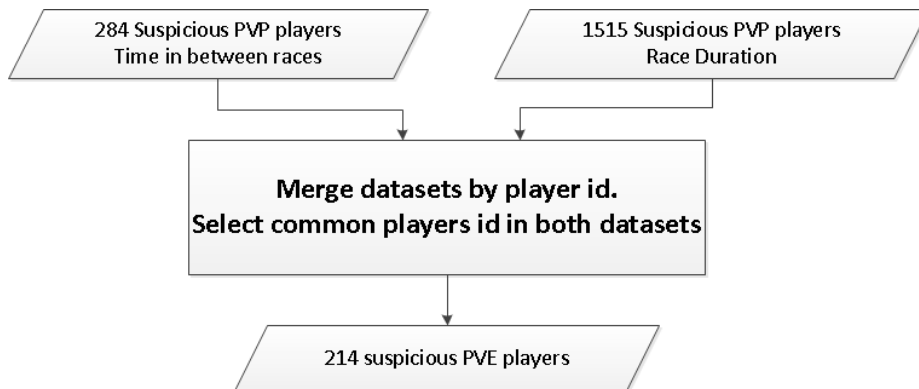


Figure 15 – PVE: Match players results

c. PPVP

A dataset with a total of 208 PPVP suspicious players is obtained after merging the two datasets from the previous steps as shown in Figure 16.

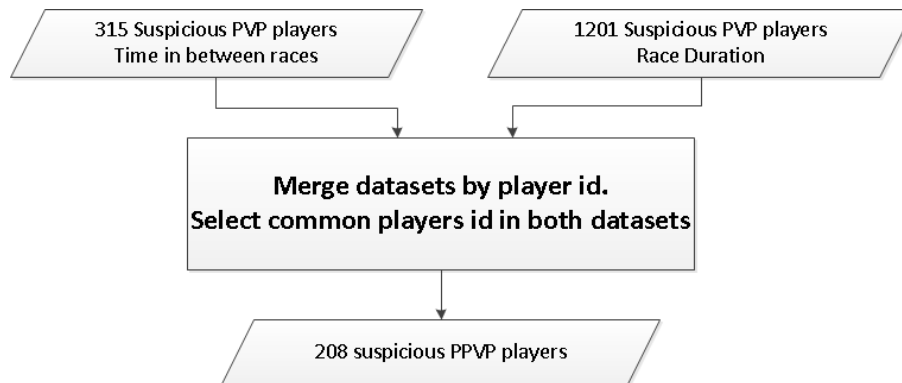


Figure 16 – PPVP: Match players results

5. Analyze players results

Players with a winning ratio greater than 70% are selected in each of the datasets obtained in step 4. A winning ratio is calculated as following:

$$\text{Winning ratio} = \frac{\text{Number of races where placing result} = 1}{\text{Total number of races}}$$

Each of the datasets is then quantitatively characterized in order to describe the potential cheaters. Figure 17 presents the resulting datasets for each mode after selecting those players with a winning ratio greater than 70%.

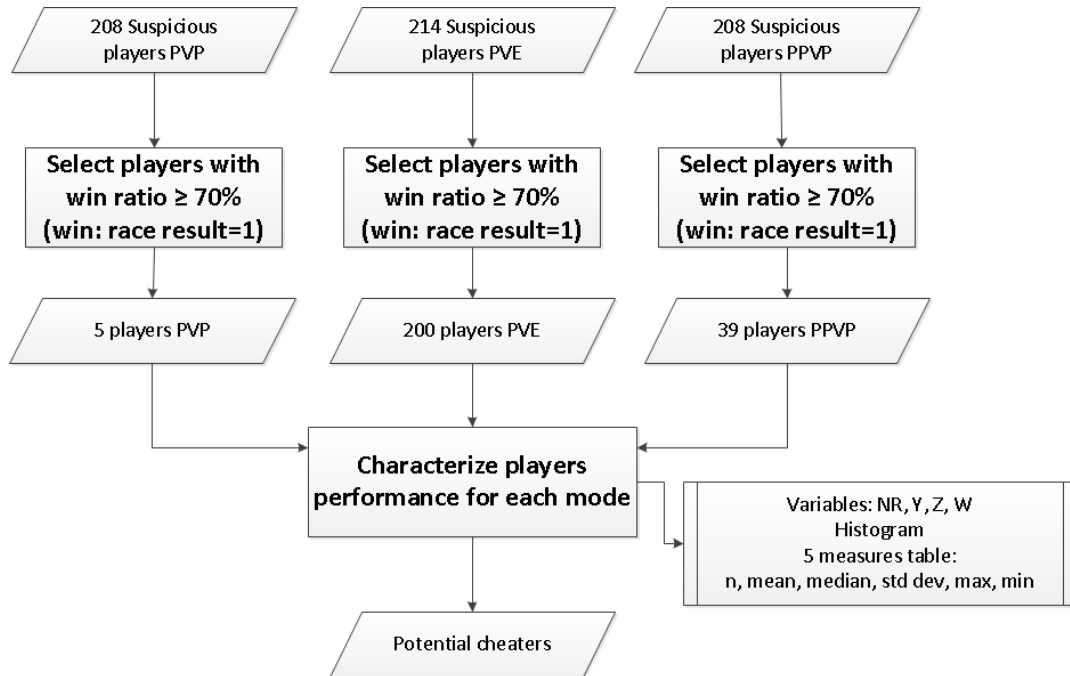


Figure 17 – Selecting players with win ratio greater than 70%

a. PVP

In PVP, five players had a winning ratio greater than 70%. Table 5 shows the listing of the five players. Table 6 shows the summary table for those players.

Table 5 – Listing of PVP potential cheaters

<i>Player ID</i>	<i>Ratio</i>	<i>SD_w</i>	<i>M_w</i>	<i>NR</i>	<i>MD_y</i>
2440167	0.89702	26.1709	97.427	4127	79.6845
2598689	0.74009	36.9491	105.583	2674	89.6870
1835770	0.72984	67.2223	154.565	2554	89.4830
2377594	0.72469	35.2953	115.467	4297	84.8230
335603	0.71611	59.2840	137.733	3769	75.8630

Table 6 – Summary table for PVP potential cheaters

<i>Variable</i>	N	Mean	Std Dev	Median	Minimum	Maximum
Ratio	5	0.7615483	0.0762269	0.7298356	0.7161051	0.8970196
Mw	5	122.1546284	23.5791950	115.4665097	97.4265040	154.5646233
SDw	5	44.9843055	17.4025575	36.9491003	26.1708825	67.2222803
NR	5	3484.20	818.0224325	3769.00	2554.00	4297.00
MDy	5	83.9081000	6.0801864	84.8230000	75.8629999	89.6869999

b. PVE

In PVE, 200 players had a winning ratio greater than 70%. Table 7 shows the listing of the top five players, and Table 8 shows the listing of the bottom five players ranked by ratio. Table 9 shows the summary table for the 200 players.

Table 7 – Top five PVE potential cheaters

<i>Player ID</i>	<i>Ratio</i>	<i>SDw</i>	<i>Mw</i>	<i>NR</i>	<i>MDy</i>
196144	0.99942	55.7540	118.584	1737	23.1670
2586349	0.99928	41.2150	127.455	1392	21.7330
20342	0.99874	40.3079	103.163	1584	19.3600
547503	0.99820	38.5989	96.318	1669	16.1170
408373	0.99786	47.1974	131.997	1400	26.8735

Table 8 – Bottom five PVE potential cheaters

<i>Player ID</i>	<i>Ratio</i>	<i>SDw</i>	<i>Mw</i>	<i>NR</i>	<i>MDy</i>
1205936	0.70431	43.6956	159.141	1045	24.5850
603416	0.70532	68.3223	153.218	1371	26.7100
1618858	0.71961	60.5630	165.328	1341	34.9370
902506	0.72386	58.2353	153.560	1463	26.6770
1140598	0.72660	70.1684	147.347	2436	28.2840

Table 9 – Summary table for PVE potential cheaters

<i>Variable</i>	N	Mean	Std Dev	Median	Minimum	Maximum
Ratio	200	0.9145127	0.0782171	0.9446342	0.7043062	0.9994243
Mw	200	130.8997602	20.5525122	127.6537506	77.2361762	177.1570000
SDw	200	52.2088048	12.9124650	52.4996310	25.9759155	115.9585874
NR	200	2365.56	1494.03	1841.00	1045.00	12107.00
MDy	200	26.6542825	6.8427806	26.0135000	11.6200001	39.8515000

A decision tree is used to characterize the data of the 200 players, and to understand their behavior within their group. In order to build the decision tree, a variable “Win” is created, and defined as follows for each player:

- Win = 1: winning ratio > 95%
- Win = 2: 90% < winning ratio ≤ 95%
- Win = 3: 80% < winning ratio ≤ 90%
- Win = 4: 70% < winning ratio ≤ 80%

The dataset was partitioned into two subsets: 70% for training, and 30% for validation. The Win variable is defined as the target variable, and the variables winning ratio (Ratio), the standard deviation of race duration (SDw), the mean of race duration (Mw), the total number of races (NR), and the median of the time in between races (MDy) are defined as the input variables for each of the 200 players. Figure 18 shows the resulting tree.

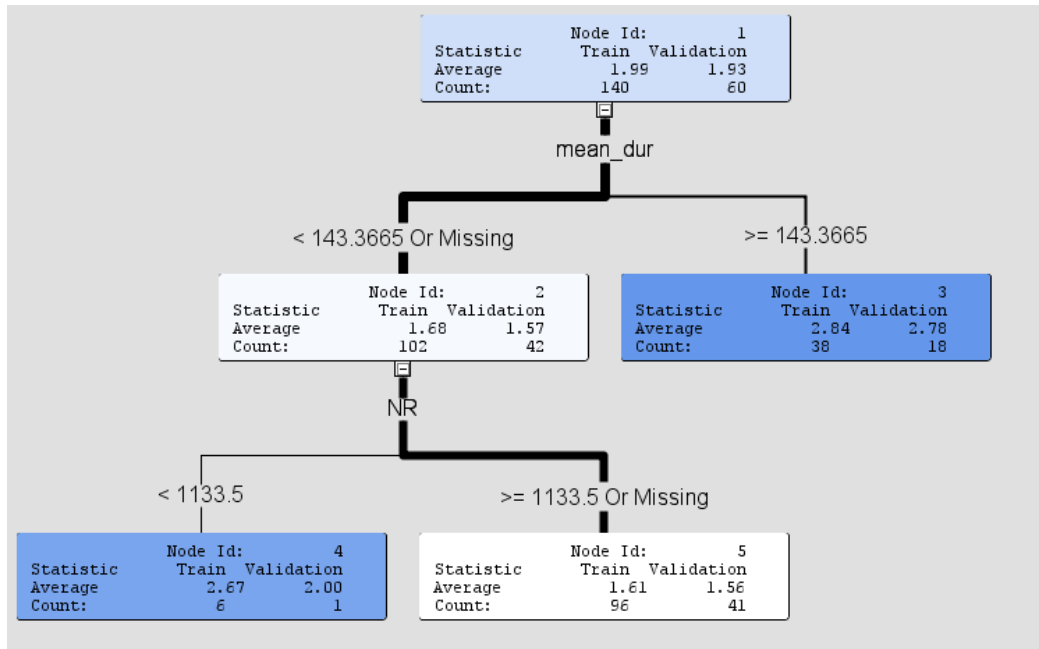


Figure 18 – PVE potential cheaters decision tree

Each of the five nodes is divided into Training and Validation. The ‘Average’ label shows the average of the target variable ‘Win’ in the training and validation dataset within the players of that node. The ‘Count’ label shows the number of players included in that node. Nodes 3, 4 and 5 represent the leaves of the tree. Each path to one of the leaves represent a decision based on the variables defined as the input variables. In this tree, only the variables mean of race duration (Mw), and the total number of races (NR) are significant in order to differentiate between the behavior of the 200 players. In the leaves, the lower the average, the more “suspicious” is the node, since that would mean that the average of the players within that node had a higher winning ratio comparing to the other nodes. In this tree, node 5 had an average of 1.61

in the training dataset, and an average of 1.56 in the validation dataset for a total of 137 players. Table 10 shows the listing of the top five players in node 5. Table 11 shows the listing of the bottom five players in node 5. Table 12 shows the summary statistics for all players in node 5.

Table 10 – Node 5: Top five Players

<i>Player ID</i>	<i>Ratio</i>	<i>SDw</i>	<i>Mw</i>	<i>NR</i>	<i>MDy</i>	<i>Win</i>
196144	0.99942	55.7540	118.584	1737	23.1670	1
2586349	0.99928	41.2150	127.455	1392	21.7330	1
20342	0.99874	40.3079	103.163	1584	19.3600	1
547503	0.99820	38.5989	96.318	1669	16.1170	1
408373	0.99786	47.1974	131.997	1400	26.8735	1

Table 11 – Node 5: Bottom five players

<i>Player ID</i>	<i>Ratio</i>	<i>SDw</i>	<i>Mw</i>	<i>NR</i>	<i>MDy</i>	<i>Win</i>
1450265	0.73556	45.8752	116.116	1195	25.2900	4
2564319	0.77947	62.5451	137.732	1578	30.4530	4
1172810	0.80748	40.0209	114.296	1418	16.1385	3
7352	0.80908	29.6146	130.442	1168	31.0600	3
1695901	0.81440	30.0699	130.134	1875	18.5200	3

Table 12 – Summary table of Node 5

<i>Variable</i>	N	Mean	Std Dev	Median	Minimum	Maximum
Ratio	137	0.9430157	0.0546877	0.9648787	0.7355649	0.9994243
Mw	137	120.2141837	12.8169479	120.1640540	77.2361762	143.3255267
SDw	137	47.9348863	9.4135371	48.7992770	25.9759155	79.3184557
NR	137	2544.04	1667.16	1864.00	1138.00	12107.00
MDy	137	25.6823650	6.8790742	25.1480000	11.6200001	39.6870000

c. PPVP

In PPVP, 39 players had a winning ratio greater than 70%. Table 13 shows the listing of the top five players, and Table 14 shows the listing of the bottom five players ranked by ratio. Table 15 shows the summary table for the 200 players.

Table 13 – Top five PPVP potential cheaters

<i>Player ID</i>	<i>Ratio</i>	<i>SDw</i>	<i>Mw</i>	<i>NR</i>	<i>MDy</i>
26535	0.98062	54.333	119.522	1187	100.990
2509004	0.97392	31.436	102.181	8129	36.008
28002	0.96797	45.431	120.487	3528	56.630
196144	0.95958	39.304	119.481	4107	38.000
2328506	0.94788	133.161	133.665	3530	43.457

Table 14 – Bottom five PPVP potential cheaters

<i>Player ID</i>	<i>Ratio</i>	<i>SDw</i>	<i>Mw</i>	<i>NR</i>	<i>MDy</i>
2031635	0.70884	53.720	120.926	735	66.1170
12832	0.70977	29.943	113.944	1013	94.1250
10403	0.70979	134.788	122.579	1144	67.3085
2394757	0.71903	53.023	136.371	1324	76.5380
1151124	0.73325	76.097	135.650	806	78.3630

Table 15 – Summary table for PPVP potential cheaters

<i>Variable</i>	N	Mean	Std Dev	Median	Minimum	Maximum
Ratio	39	0.8231787	0.0846312	0.8150209	0.7088435	0.9806234
Mw	39	123.4107353	11.6565242	121.1429877	102.1814303	151.7804565
SDw	39	55.8799174	24.8484473	51.4795306	26.0753423	134.7882938
NR	39	1554.97	1551.66	956.0000000	424.0000000	8129.00
MDy	39	72.0610897	17.8246884	69.2185000	36.0085000	101.7284999

A decision tree is used to characterize the data of the 39 players, and to understand their behavior within their group. In order to build the decision tree, a binary variable “Win” is created, and defined as follows for each player:

- Win = 1: winning ratio > 90%
- Win = 0: 70% < winning ratio ≤ 90%

The dataset was partitioned into two subsets: 70% for training, and 30% for validation. The Win variable is defined as a binary target variable, and the variables winning ratio (Ratio), the standard deviation of race duration (SDw), the mean of race duration (Mw), the total number of races (NR), and the median of the time in between races (MDy) are defined as the input variables for each of the 39 players. Figure 19 shows the resulting tree.

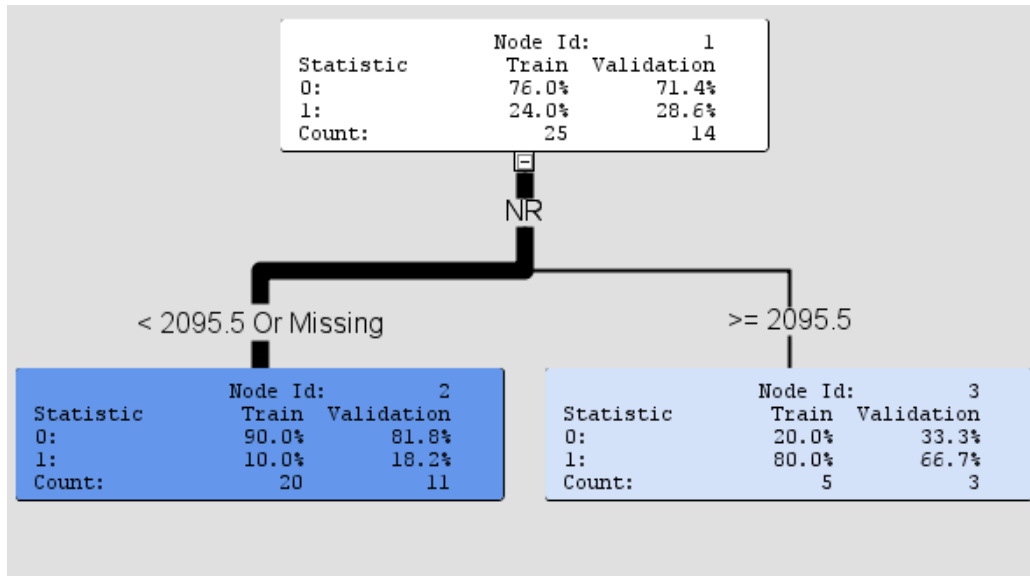


Figure 19 – PPVP potential cheaters decision tree

Each of the three nodes is divided into Training and Validation. The percentage of observations with a value of zero or one is indicated with the labels ‘0:’ and ‘1:’ in the training and validation dataset. The ‘Count’ label shows the number of players included in the training and validation subset. Nodes 2 and 3 represent the leaves of the tree. Each path to one of the leaves represent a decision based on the variables defined as the input variables. In this tree, only the variable of the total number of races (NR) is significant in order to differentiate between the behavior of the 39 players. Node 3 is the more “suspicious” node because the number of 1’s for the Win variable is higher comparing to the other leaves (80% in the training set, and 66.7% in the validation set). Table 16 shows the listing for all the players in node 3. Table 17 shows the summary statistics for all players in node 5.

Table 16 – Node 3: Listing of players

<i>Player ID</i>	<i>Ratio</i>	<i>SDw</i>	<i>Mw</i>	<i>NR</i>	<i>MDy</i>	<i>Win</i>
2509004	0.97392	31.436	102.181	8129	36.0085	1
28002	0.96797	45.431	120.487	3528	56.6300	1
196144	0.95958	39.304	119.481	4107	38.0000	1
2328506	0.94788	133.161	133.665	3530	43.4570	1
1221618	0.91339	53.108	125.098	2367	68.9995	1
2846948	0.90612	37.233	122.167	4133	61.260	1
2197688	0.82380	44.736	124.104	4387	101.728	0
1000564	0.81502	44.750	117.478	2157	100.691	0

Table 17 – Summary table of Node 3

<i>Variable</i>	<i>N</i>	<i>Mean</i>	<i>Std Dev</i>	<i>Median</i>	<i>Minimum</i>	<i>Maximum</i>
Ratio	8	0.9134600	0.0629307	0.9306339	0.8150209	0.9739205
Mw	8	120.5827768	8.9116146	121.3270541	102.1814303	133.6654302
SDw	8	53.6450412	32.7703637	44.7433764	31.4363195	133.1610716
NR	8	4042.25	1840.82	3818.50	2157.00	8129.00
MDy	8	63.3468750	26.0072839	58.9450001	36.0085000	101.7284999

5. DISCUSSION AND FUTURE WORK

In this work, we have presented a methodology to detect bot cheaters in a MMO racing game. Bots perform certain repetitive or precise tasks in place of human gamers [11]. Our methodology builds rules that allow us to identify suspicious players. These rules were based on empirical values after examination of graphs and summary statistics along with expected cheater behavior. A player is considered suspicious of using bots in the game if they present a high number of races played with race duration showing low mean and low standard deviation, and time in between races showing consistent low median value.

In our case study, evidence suggests that bot cheating is present largely in PVE mode of racing, where automation is most effective because of deterministic behavior.

Examination of the summary statistics tables for the suspicious players in each mode, revealed only 5 PVP players versus 200 PVE players and 39 PPVP players with a winning ratio greater than 70%. Furthermore, in PVE mode, the median of time in between races is lower (MDy=26.65), and the mean winning ratio is higher (Ratio=0.91) than in PVP (MDy=83.90, Ratio=0.76) or PPVP (MDy=72.06, Ratio=0.82). Similar values for the standard deviation of the median time in between races (6.08 sec. for PVP potential cheaters, 6.84 sec. for potential PVE cheaters and 6.87 sec. for node 5 of the decision tree for PVE potential cheaters) may be evidence of an automated start after a race ends, which might be an indication of the use of bots in racing. A consistent repetitive movement characterizes bots, and constant standard deviation could be an indicator of the use of bots.

Inspection of the top five PVE potential cheaters revealed that the winning ratio for these players is between 99.78% (NR=1400) and 99.94% (NR=1737). This is equivalent to only 4

or less races in which the player did not score first place in the races. Inspection of the PVP potential cheaters revealed that, even though the total number of races played by player is between 2554 and 4297 races, the winning ratio in PVP, between 71.61% and 89.70%, is less than in PVE. Similarly, examination of the top five PPVP potential cheaters revealed that the winning ratio for these players is between 94.79% and 98.06%, with a total number of races played by player between 1187 and 8129.

A decision tree has been presented in this work for the PVE and PPVP mode based on the metrics of the players with a winning ratio greater than 70%. The decision tree provides rules to characterize data into groups. In the decision tree, only the significant variables are used to build the decision rules. In the PVE tree, the mean of race duration (Mw), and the total number of races played (NR) are significant in order to differentiate between the behavior of the 200 players. In the other side, in the PPVP tree, only the number of races played (NR) was needed to differentiate between the players.

Future work includes building a real-time tool that will detect and flag suspicious player behavior. In this work, we have presented cutoff values based after examination of graphs and summary statistics along with expected cheater behavior. However, different values for the cutoff values can be tried as a next step. The tool will include the exploration of different cutoff values that will allow for more aggressive cheater detection.

REFERENCES

- [1] ESA Entertainment Software Association. “The Entertainment Software Association - Sales & Genre Data.” Internet: <http://www.theesa.com/facts/salesandgenre.asp>, [Nov. 06, 2012].
- [2] S. Siwek. Videogames in the 21st Century. http://www.theesa.com/facts/pdfs/VideoGames21stCentury_2010.pdf.
- [3] Strategy Analytics. “Online Game Revenue Fuels Global Video Game Software Market.” Internet: <http://www.strategyanalytics.com/default.aspx?mod=pressreleaseviewer&a0=4862>, Feb. 19, 2000 [Nov. 06, 2012].
- [4] C. Steinkuehler. Learning in massively multiplayer online games. In *Proceedings of the Sixth International Conference of the Learning Sciences*, ICLS '04, pages 521-528, Santa Monica, CA, USA, 2004. ACM.
- [5] J. Yan and B. Randell. An investigation of cheating in online games. *Security Privacy, IEEE*, 7(3):37–44, may-june 2009.
- [6] P. Laurens, R. F. Paige, P. J. Brooke, and H. Chivers. A novel approach to the detection of cheating in multiplayer online games. In *Proceedings of the 12th IEEE International Conference on Engineering Complex Computer Systems, ICECCS '07*, pages 97–106, Washington, DC, USA, 2007. IEEE Computer Society.
- [7] L. Galli, D. Loiacono, L. Cardamone, and P. Lanzi. A cheating detection framework for unreal tournament iii: A machine learning approach. In *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, pages 266 –272, 31 2011-sept. 3 2011.
- [8] L. Chapel, D. Botvich, and D. Malone. Probabilistic approaches to cheating detection in online games. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 195–201, aug. 2010.
- [9] H.-K. Pao, K.-T. Chen, and H.-C. Chang. Game bot detection via avatar trajectory analysis. *Computational Intelligence and AI in Games, IEEE Transactions on*, 2(3):162 –175, sept. 2010.

- [10] H. Kim, S. Hong, and J. Kim, "Detection of auto programs for MMORPGs," in Proceedings of the 18th Australian Joint Conference on Artificial Intelligence (AI '05) on, pages 1281–1284, Sydney, Australia, dec. 2005.
- [11] D. Bethea, R. Cochran, and M. Reiter, "Server-Side Verification of Client Behavior in Online Games," in Proceedings of the 17th Annual Network and Distributed System Security Symposium of the Internet Society on, pages 32:1--32:27, San Diego, California, dec. 2010.